

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A problem solving method for use in a computer system, wherein the computer system includes an applications module having a problem solver, wherein the problem solver comprises a solving module and a complexity module, the method comprising ~~the steps of:~~

receiving a problem statement from the applications module;

predicting an expected problem solver behavior associated with said configuration parameters for said problem statement;

providing the solving module with said configuration parameters;

selecting a set of configuration parameter vectors;

determining a set of search space points;

perform a partial search with said configuration parameter vectors;

determining actual solver behavior;

reviewing said actual solver behavior incrementally to determine if a problem solution has been found, wherein reviewing comprises comparing said expected solver behavior with said actual solver behavior ;

determining whether to perform a solver iteration step or to request the complexity module to perform an adaptation step if a problem solution has not been found, wherein said complexity module includes means for detecting problem complexity, means for estimating rate of improvement bounds, and means for selecting a complexity related action, wherein said complexity related action comprises selecting at

least one member from the group consisting of returning an error, accepting the best result possible for fixed computational resources, adding computational resources, and changing the problem statement;

performing a said solver iteration step when said solver iteration step is selected, comprising the steps of determining new actual solver behavior and determining whether to repeat said solver iteration step;

repeating said solver iteration step until said adaptation step is selected;

comparing said actual solver behavior with said expected solver behavior when said adaptation step is selected;

requesting the complexity module to perform said adaptation step;

performing said adaptation step, comprising the steps of modifying said configuration parameters within the complexity module, configuring the solving module with said modified configuration parameters, determining expected solver behavior associated with said modified configuration parameters for said problem statement, selecting an algorithm to calculate a revised problem solution, determining a revised actual solver behavior associated with said modified configuration parameters for said problem statement, reviewing said revised actual solver behavior to determine if a problem solution has been found, determining whether to perform said solver iteration step or to request the complexity module to perform another adaptation step if a problem solution has not been found, and repeating said iteration step until said adaptation step is selected;

repeating said adaptation step until a problem solution is found; and

providing the solution to the applications module.

2. (Original) The problem solving method according to claim 1, further comprising the step of selecting an algorithm to calculate an initial problem solution.

3. (Previously Presented) The problem solving method according to claim 1, further comprising the step of refining the configuration parameters provided to the solving model.
4. (Original) The problem solving method according to claim 1, wherein the problem solver comprises an adaptive constraint problem solver.
5. (Original) The problem solving method according to claim 1, further comprising the step of transforming said problem statement after receiving said problem statement from the applications module.
6. (Canceled)
7. (Original) The problem solving method according to claim 1, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.
8. (Original) The problem solving method according to claim 7, further comprising the step of transforming said problem configuration parameters before providing said problem configuration parameters to the solving module.
9. (Original) The problem solving method according to claim 1, wherein the step of selecting a set of configuration parameter vectors further comprises:
 - choosing a set of default configuration parameter vectors;
 - selecting an initial minimum point;
 - performing a local search;
 - evaluating actual behavior to determine whether to repeat a local search or select a different solver algorithm;
 - repeating a local search with a second minimum point when the step of repeating a local search is selected; and

revising the set of configuration parameter vectors for each search performed.

10. (Currently Amended) A computer system for problem solving, the system having implementation units communicating with the computer system, the system comprising:

an input device for providing the problem statement;

a computer coupled to the output of said input device;

a memory portion coupled to the computer comprising:

software for receiving the problem statement from said input device;

software for determining solver configuration parameter vectors;

software for configuring a problem solver, wherein said problem solver includes a solving module and a complexity module, wherein said complexity module includes means for detecting problem complexity, means for estimating rate of improvement bounds, and means for selecting a complexity related action, wherein said complexity related action comprises selecting at least one member from the group consisting of returning an error, accepting the best result possible for fixed computational resources, adding computational resources, and changing the problem statement;

software for predicting expected solver behavior;

software for performing a partial search with said configuration parameter vectors;

software for determining actual solver behavior and incrementally determining whether a solution has been found, wherein determining whether a solution has been found comprises comparing said expected solver behavior and said actual solver behavior;

software for determining whether to perform a solver iteration step or to perform an adaptation step; and

software for performing an adaptation step, comprising modifying said configuration parameters and reconfiguring said problem solver; and

output means for providing a solution statement.

11. (Original) The computer system for problem solving according to claim 10, wherein said problem solver comprises an adaptive constraint problem solver.

12. (Original) The computer system for problem solving according to claim 10, wherein said memory portion further comprises software including a learning module for refining said expected problem solver behavior.

13. (Original) The computer system for problem solving according to claim 10, further comprising a problem transformer module for transforming said problem statement after receiving said problem statement from said input device.

14. (Canceled)

15. (Original) The computer system for problem solving according to claim 10, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

16. (Original) The computer system for problem solving according to claim 15, further comprising a problem transformer module for transforming said problem configuration parameters before providing said problem configuration parameters to said solving module.

17. (Original) The computer system for problem solving according to claim 10, wherein said software for determining expected solver behavior comprises a data structure, said data structure containing configuration parameters and expected behaviors for a plurality of problem types.

18. (Previously Presented) The computer system for problem solving according to claim 10, wherein the computer includes an embedded computer.

19. (Original) The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a copier or printer.

20. (Original) The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a process control system.

21. (Original) The computer system for problem solving according to claim 18, wherein said embedded computer system controls at least one operation within a diagnostics unit.

22. (Currently Amended) A computer system for problem solving, the system having implementation units communicating with the computer system, the system comprising:

an input device for providing the primary goal for the task to be performed;

a computer coupled to the output of said input device;

a memory portion coupled to said computer comprising:

a complexity module for configuring a problem statement and predicting expected solver behavior, wherein said complexity module includes means for detecting problem complexity, means for estimating rate of improvement bounds, and means for selecting a complexity related action, wherein said complexity related action comprises selecting at least one member from the group consisting of returning an error, accepting the best result possible for fixed computational resources, adding computational resources, and changing the problem statement;

a controllable solving module coupled to said complexity module for determining actual solver behavior;

a synthesis module for determining configuration parameter vectors; and

comparison means for comparing said actual solver behavior with said expected solver behavior and determining whether a problem solution has been found; and

output means for providing a statement of the problem solution.

23. (Original) The computer system for problem solving according to claim 22, wherein said problem solver comprises an adaptive constraint problem solver.

24. (Original) The computer system for problem solving according to claim 22, further comprising a learning module.

25. (Original) The computer system for problem solving according to claim 22, further comprising a problem transformer module for transforming said problem statement after receiving said problem statement from said input device.

26. (Canceled)

27. (Original) The computer system for problem solving according to claim 22, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

28. (Original) The computer system for problem solving according to claim 27, further comprising a problem transformer module for transforming said problem configuration parameters before providing said problem configuration parameters to said solving module.

29. (Original) The computer system for problem solving according to claim 22, wherein said complexity module comprises a data structure, said data structure containing configuration parameters and expected behaviors for a plurality of problem types.

30. (Original) The computer system for problem solving according to claim 22, wherein said control computer comprises an embedded computer system.

31. (Original) The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a copier or printer.

32. (Original) The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a process control system.

33. (Original) The computer system for problem solving according to claim 30, wherein said embedded computer system controls at least one operation within a diagnostics unit.

34. (Currently Amended) A problem solver in the form of a program to be run on a computer system, said problem solver comprising:

means for receiving a problem statement;

means for detecting problem complexity;

means for estimating rate of improvement bounds;

means for selecting a complexity related action, wherein said complexity related action comprises selecting at least one member from the group consisting of returning an error, accepting the best result possible for fixed computational resources, adding computational resources, and changing the problem statement;

means for predicting expected solver behavior associated with said problem statement;

means for providing configuration parameters for a plurality of problems;

means for determining a set of configuration parameter vectors;

means for performing a partial search with said configuration parameter vectors;

means for calculating actual solver behavior;

means for reviewing said actual solver behavior incrementally to determine if a problem solution has been found, wherein reviewing comprises comparing said expected solver behavior to said actual solver behavior;

means for determining whether to perform a solver iteration step or to request an adaptation step if a problem solution has not been found;

means for performing a solver iteration step, comprising performing another search step, calculating a revised actual solver behavior and determining whether to repeat said solver iteration step;

means for comparing said actual solver behavior with said expected solver behavior;

means for requesting performance of an adaptation step;

means for performing an adaptation step, comprising modifying said configuration parameters, determining a revised expected problem solver behavior, and providing said modified configuration parameters and said revised expected problem solver behavior to said means for performing a solver iteration step; and

means for providing the problem solution to an output device.

35. (Currently Amended) A problem solving method in the form of a program to be run on a computer system comprising ~~the steps of:~~

receiving a problem statement;

means for detecting problem complexity;

configuring a problem solver with configuration parameters;

determining a set of configuration parameter vectors;

predicting expected solver behavior associated with said configuration parameters for said problem statement;

searching for a solution with said configuration parameter vectors;

determining actual solver behavior;

means for estimating rate of improvement bounds;

determining if a problem solution has been found, comprising comparing said expected solver behavior with said actual solver behavior incrementally;

means for selecting a complexity related action, wherein said complexity related action comprises selecting at least one member from the group consisting of returning an error, accepting the best result possible for fixed computational resources, adding computational resources, and changing the problem statement;

determining whether to perform a solving iteration step or an adaptation step if a problem solution has not been found;

performing said solver iteration step, when said solver iteration step is selected, comprising the steps of determining a new actual solver behavior and determining whether to repeat said iteration step;

repeating said solver iteration step until said adaptation step is selected;

comparing said actual solver behavior with said expected solver behavior when said adaptation step is selected;

performing said adaptation step, comprising the steps of modifying said configuration parameters, determining expected solver behavior associated with said modified configuration parameters, determining a revised actual solver behavior, reviewing said revised actual solver behavior to determine if a problem solution has been found, determining whether to perform said solver iteration step or to perform another adaptation step if a problem solution has not been found, and repeating said iteration step until said adaptation step is selected;

repeating said adaptation step until a problem solution is found; and

transmitting a solution statement.

36. (Original) The problem solving method according to claim 35, wherein said problem solving method comprises an adaptive constraint problem solving method.

37. (Previously Presented) The problem solving method according to claim 35, further comprising the step of refining the configuration parameters.

38. (Previously Presented) The problem solving method according to claim 35, further comprising the step of transforming said problem statement.

39. (Canceled)

40. (Previously Presented) The problem solving method according to claim 35, wherein said configuration parameters include problem configuration parameters and solver configuration parameters.

41. (Previously Presented) The problem solving method according to claim 40, further comprising the step of transforming said problem configuration parameters.

42. (Previously Presented) The problem solving method according to claim 35, wherein the step of determining a set of configuration parameter vectors further comprises:

choosing a set of default configuration parameter vectors;

selecting an initial minimum point;

performing a local search;

evaluating actual behavior to determine whether to repeat a local search or select a different solver algorithm;

repeating a local search with a second minimum point when the step of repeating a local search is selected; and

revising the set of configuration parameter vectors for each search performed.

43. (Previously Presented) The problem solving method according to claim 35, further comprising the step of selecting an algorithm for calculating a problem solution.

44. (Canceled)

45. (Currently Amended) An article of manufacture comprising a computer usable medium having computer readable program code embodied in said medium which, when said program code is executed by said computer causes said computer to perform method steps for problem solving for use in a computer system, wherein the problem solver includes a solving module and a complexity module, said method comprising:

receiving a problem statement from the applications module;

predicting an expected problem solver behavior associated with configuration parameters for said problem statement;

providing the solving module with said configuration parameters;

selecting a set of configuration parameter vectors;

determining a set of search space points;

perform a partial search with said configuration parameter vectors;

determining actual solver behavior;

reviewing said actual solver behavior incrementally to determine if a problem solution has been found, wherein reviewing comprises comparing said expected solver behavior with said actual solver behavior;

determining whether to perform a solver iteration step or to request the complexity module to perform an adaptation step if a problem solution has not been found, wherein said complexity module includes means for detecting problem complexity, means for estimating rate of improvement bounds, and means for selecting a complexity related action, wherein said complexity related action comprises selecting at least one member from the group consisting of returning an error, accepting the best

result possible for fixed computational resources, adding computational resources, and changing the problem statement;

performing a said solver iteration step when said solver iteration step is selected, comprising the steps of determining new actual solver behavior and determining whether to repeat said solver iteration step;

repeating said solver iteration step until said adaptation step is selected;

comparing said actual solver behavior with said expected solver behavior when said adaptation step is selected;

requesting the complexity module to perform said adaptation step;

performing said adaptation step, comprising the steps of modifying said configuration parameters within the complexity module, configuring the solving module with said modified configuration parameters, determining expected solver behavior associated with said modified configuration parameters for said problem statement, selecting an algorithm to calculate a revised problem solution, determining a revised actual solver behavior associated with said modified configuration parameters for said problem statement, reviewing said revised actual solver behavior to determine if a problem solution has been found, determining whether to perform said solver iteration step or to request the complexity module to perform another adaptation step if a problem solution has not been found, and repeating said iteration step until said adaptation step is selected;

repeating said adaptation step until a problem solution is found; and

providing the solution to the applications module.